The State of AI & Engineering in 2026

AI has transitioned from pilot to production. Here's what engineers, product managers, designers and technology leaders in Australia told us.

CONTENTS

01 Why we did this
02 Executive Summary
03 What we heard
04 The benefits are real
05 Key structural changes AI is driving
06 Reshaping roles
07 Reshaping hiring
08 The downsides
09 What's old is new again
10 Thank you from Lookahead
11 Appendix

§1 Why we did this

It feels like we're living in dog years.

So much change, all at once. Opinions are divided on what it means for the way we work but what's clear is that we're living in a once-in-a-career moment.

At Lookahead, we wanted to make something useful for the community as we made sense of the changes ourselves. Putting this together helped us learn, we hope you learn something too.

We started by interviewing a few dozen leaders in Australia and the US. Some aren't named because of who their employer is, but they're at the frontier and have helped us a lot. We then went to the wider community via our survey. This report exists because people were generous with their time and honest about their experiences, so we're making it openly available in that same spirit.

Thanks for telling us how you're feeling. We're both energised and tired after what has already been a big year. We're long term optimistic and feel motivated to run hard for a bit. It's nice to know we're not alone.



Steve Gilles
Founder, Lookahead

THANK YOU

~300 people completed the survey or conducted an interview. There was so much generosity in time and spirit. Survey results are anonymised so please trust us when we say a big thanks to the many people who wrote long thoughtful free text answers. They shaped this report as much as the numbers did. Also thank you to the engineering and technology leaders below for helping us shape this report. Finally, a huge thanks to the Lookahead team who all pitched in to make this a reality - Fiona Chan, Debbie Teakle, Sarah Jacob, Kyle Jackson and Christopher Concepcion. Took a lot of work to figure all this out, run interviews, gather survey responses, give feedback and make edits.



Alex North
independent, formerly Google



Alexandra Swan
Software Engineer & Product Manager, Ajust



Carl Woodward
CTO, Ordermentum



Chaitanya Kuber
CTO, Airtasker



Ciaran Hale
CTO, Deputy



Darren Smith
Senior Director Product Design, Multiplier



Dave Slutzkin
Cofounder and CEO, Cadence



Dave Tong
Cofounder and CPTO, Employment Hero



Gareth Stokes
Cofounder at Super IT



Graham Darby
Data & AI Platform Manager, Tyro Payments



James Brett
CTO, Slatterys



John Allsopp
Founder, Web Directions



Manik Surtani
CTO and Cofounder, Agentic AI Foundation



Mic Neale
Principal Engineer, Applied AI, Block



Mike Farah
Principal Engineer, Contact Harald, formerly Atlassian



Oz Nova
Founder, CS Primer and Bradfield School of Computer Science



Paul Keen
VP of Engineering, Culture Amp



Pierre Bergamin
CTO, InDebted



Ray Tung
VP of Engineering, Kasada



Shaun Domingo
CTO, Macquarie Cloud Services



Tim Lucas
Cofounder, Amp



Vanessa Ung
Founding Engineer, Lorikeet



Wendy Glasgow
formerly CTO GROW Inc, now HotDoc

§2 Executive Summary

It was a good year that wore people out. 80% of the 272 people who answered our survey said AI's impact on their role over the past year was positive, and only 7% said negative. Ask for a single word instead of a rating and the picture changes: "excited" and "empowered" top the list, but "exhausted", "tired" and "stressed" sit right behind them, and only 43% of the words people chose were positive. Shipping got faster for 92% of respondents. Workload and pressure went up for 68%. Job satisfaction tilted the other way: 44% said it fell, 35% said it rose, and 57% saw no change in pay.

AI is no longer optional. Every leader we interviewed said, one way or another, that refusing to use AI is now disqualifying; the comparison we heard more than once was refusing to use an IDE. The gains are real. Organisations told us about three times the output per engineer, a six-month project delivered in two, an incident diagnosed in seven minutes that would once have taken four hours. What changes most is what's possible. The default answer to a new idea is no longer "no", it's "let me check", and work that used to die at the bottom of the backlog now gets done. Quality is the exception: 45% say it fell against 33% who say it improved, and time spent in code review rose for 57%. We have made code cheap to write and have not yet made it cheap to check.

How software gets built is being rethought, quickly. Non-engineers are shipping working prototypes as pull requests; not one respondent told us they were seeing no cross-role movement. People in compliance, finance and legal are building their own tools. Leaders are back on the tools too: 51% of engineering leaders are more hands-on than a year ago, and 23% much more. Core engineering teams are getting smaller, but we don't read that as less demand for engineers. Our view is that engineers get dispersed across the organisation, into finance, ops and marketing, as the adult supervision for the custom software every department can now build for itself. Ownership of production has not moved, though. Every leader we spoke to drew the same line, and the product managers who tried to cross it pulled back once they realised that shipping meant being on call.

Juniors are the industry's open question. 17 of our 22 long-form interviews raised it without being asked. The entry-level work that trained a generation of engineers now sits below what a model does unsupervised, and nobody we spoke to could describe the replacement on-ramp. The optimists think juniors will learn faster precisely because they aren't typing. The early US labour-market data, a roughly 19% relative employment gap for 22–25 year olds in AI-exposed occupations, sides with the pessimists for now. Our sample has a median of 17 years in the industry, so every account of the junior experience here is a senior person's account of it.

Tokens became a management question this year, and most teams haven't answered it. The most common budget is $150–500 per engineer per month, and 14% of respondents don't know theirs. Leaders believe usage is carefully managed; the people they lead don't see it that way. Most of the leaders we interviewed think quotas are the wrong lever altogether, and several warned that the heaviest token users are not the best AI users. The tension worth watching is that in some organisations the token budget and the headcount budget now come from the same bucket. It isn't zero-sum, but it is no longer separate.

The fear is deskilling, not displacement. 35% named over-reliance and skill atrophy as AI's most significant downside. Job security got 10%. 13 of 22 interviews described something closer to mourning: a craft built over a career, changed in a year. The job now rewards supervising several streams of machine output at once rather than long stretches of single-task flow, and that taxes exactly the people who were best at the old way of working.

Hiring is turning into an audition. 60% of organisations have changed their interview process because of AI, and almost all of that change landed on the take-home and the live coding round. The emerging pattern is an hour in an unfamiliar codebase with your own tools, judged on how you frame the problem and critique what the model produced rather than on the finished solution. The market is tight. Only 4% of engineering leaders are hiring more engineers and a quarter are hiring fewer, but the premium for exceptional engineers is growing.

The skills that grew are the ones that were always hard to teach. Product thinking (55%) and system design (52%) sit clear of everything else. Coding itself and deep knowledge of a particular stack are what respondents volunteered as declining.

A note on who told us this. 272 survey responses and 22 interviews, fielded through Lookahead's network: Sydney-weighted, 86% engineers, senior, and not random. Findings are directional, every exhibit states its own n, and where a comparison rests on a subgroup we've printed the statistical grade beside it.

Other research you can check out. Our sample data is specific to Australia so that's valuable, but there are reports with bigger sample sizes. Lenny's is how tech workers are feeling in 2026 is one and it will be interesting to see the results of Stack Overflow's 2026 report when it lands.

§3

What we heard

80%

positive about AI's impact on their role

39%

chose the strongest option on the scale

8%

were negative about the year

Only 8% selected "negative" or "very negative", but look at the most common words chosen to describe the year - excited, empowered, exhausted and stressed. It's been a year of type 2 fun.



Steve Gilles
Founder, Lookahead

AI isn't just evolving; it's sprinting - at a velocity unseen in any past shift. It's fun to realise the Stack Overflow's 2025 Developer Survey is now woefully out of date as it closed in June 2025, before Opus 4.5. Still, back then 84% of developers were already weaving AI into their daily workflow. 2026's survey will be closer to 100% and "weaving" will be an understatement.

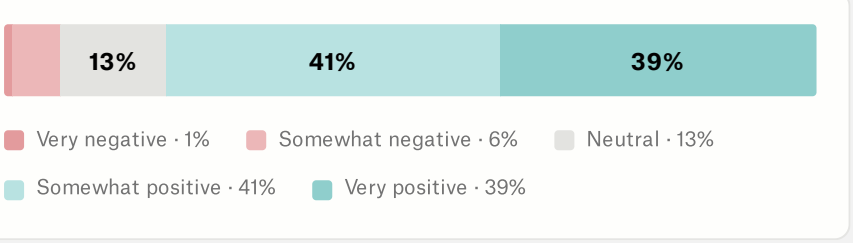
Creating quickly feels amazing, yet beneath the surface, it's fuelling a new breed of fatigue and pressure that we are only just beginning to grasp.

80% of respondents told us AI's impact on their role over the past year was positive, and 39% were very positive. Only 7% were outright negative.

EXHIBIT 3.1

More than four in five said the year was positive, and the top box was the most popular answer on the scale.

Overall, AI's impact on your role this past year has been... · n=271 · mean 4.10 of 5



n=271 · survey of 272 responses

Only four respondents in 271 chose "very negative".

"There's AI years and there's normal years. There's an AI week, which is a normal year."



Tim Lucas
Cofounder, Amp

"From when I got my first computer in about 1981, 1982 - this is without doubt the most impactful year of my career. Without doubt I have been more productive. I have had more professional growth. I think I probably in some respects had more personal growth, and I don't think they're unrelated."



John Allsopp
Founder, Web Directions

For senior leadership, the primary observation is the speed of change.

There are also those who are feeling incredibly energised and empowered by the changes afoot.

Some who have 'seen this before' are less impressed

Some people felt like they have watched this unfold before; cloud, mobile, Web 2.0 etc and that each wave felt existential when inside it. The pattern they recognise is: a platform shift arrives, everyone re-tools, the roles reshuffle, and five years later it's just how software gets built.

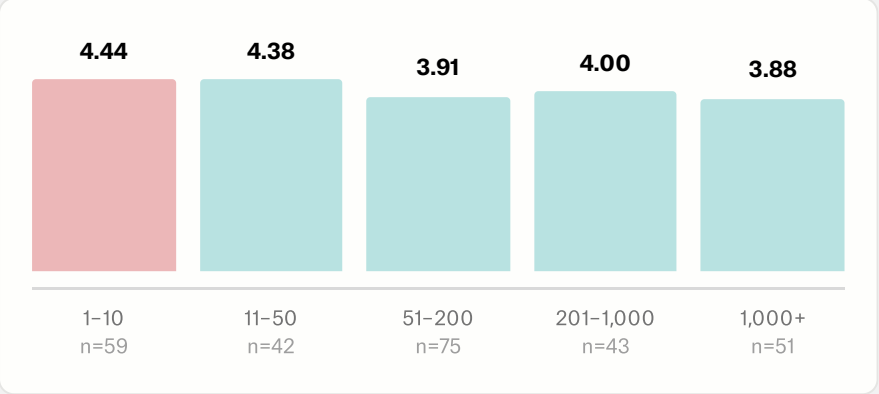
So who is feeling optimistic?

It might be easy to assume that the temptation of increased output makes managers and executives the enthusiasts and individual contributors sceptics but the data shows company size matters most.

EXHIBIT 3.2

People at the smallest companies had a markedly better year than people in the middle.

Mean overall impact by company size · n=270 · trend ρ −0.25 Solid q=0.013



n=270 · survey of 272 responses

The same gradient shows up on a combined index of seven separate questions, so this isn't one question behaving oddly. With the full sample in, this finding now clears our strictest statistical bar.

This is not to say that seniority does not have an effect. C-suite respondents averaged 4.58 against 3.91 for individual contributors - but when we tested tenure, seniority and company size in the same model, **company size mattered far more**. Those in a sub ten-person company predicted a far more positive impact of AI than those in larger companies. In the combined model, company size carried the most weight (β −0.20, p=0.001); a seniority effect held on (β +0.17, p=0.008), and tenure added nothing once both were in.

> "AI is a significant shift, but it is part of a familiar pattern in technology: new capabilities emerge quickly, and the real work is learning where they genuinely create value."



Ciaran Hale
CTO, Deputy

> "It feels like this is what it felt like for those who were around in the 2005 sort of era when it was all Web 2.0 and we were learning about Ajax... everyone was really excited and really passionate... feels like that now, with AI."



Gareth Stokes
Cofounder, Super IT

There is a caveat that this raises. Are people who are bullish about AI also more likely to join smaller fast-moving companies? Research suggests self-selection of this kind is real: a Management Science study of first-time employment choices (Roach & Sauermann) found people who prefer startup work show measurably higher risk tolerance, stronger preference for autonomy and greater interest in new technology than peers who prefer established firms - so some of the gradient in Exhibit 3.2 is likely people sorting themselves, not company size doing the work.

The words people chose

At the end of the survey we asked an optional question: in one word, how has AI made you feel about your work? 186 people answered, and the answers do not mirror our headline 80% positive result.

excited the most common single answer, from 11 people

empowered 8 · the late responses pushed it off the top spot

exhausted 5 · followed by tired, 4 · optimistic also 5

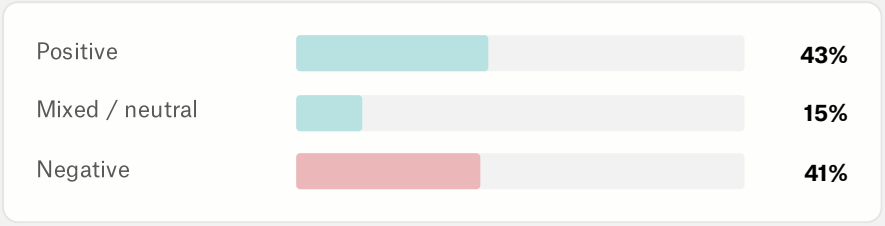
stressed 3 · alongside anxious, 4, and hectic, rushed, scared and cautious, 2 each

Somewhat alarmingly, when we do some basic semantic analysis of the words used, the world is not all that rosy.

EXHIBIT 3.3

Classified by sentiment, the one-word answers still lean far less positive than the ratings.

Our classification of the 184 usable answers (two test entries excluded) · full word list in the appendix



n=184 · 2 test entries excluded · survey of 272 responses

Borderline words move the split by only a point or two whichever way they're classified. The late responses skewed positive - the mid-survey data was a dead heat - but the gap between asking for a rating (80% positive) and asking for a feeling (43% positive) remains the point.

The last 12 months have clearly been a year that most people found beneficial but at the same time found exhausting.

KEY IMPLICATION

The enthusiasm is genuine but it is not uniform. Four in five people had a positive experience but the people having the best experience are at the smallest companies and in management roles. Beyond the hype though are some underlying concerns that we as an industry need to be aware of.

This is an interesting finding. We're definitely seeing teams get smaller, so it's nice to know that if we're headed that way, the folks already there are largely happy about it. In my experience smaller teams have often reaped the benefits of change but this doesn't mean all teams can be small.

When an IC can get so much done paired with an LLM, building the right thing in the first place, and having tight feedback loops is more important. Code being slow gave a business time to think long and hard about priorities.

Smaller teams doesn't need to mean fewer developers being hired. Engineers will be dispersed across organisations, rather than being in the silo of a dedicated product team. Take talent engineering as an example - this role didn't exist in 2025 and now you'll find it in scaleups. Heck, Lookahead has one. This year we started running our own stack and now spend an order of magnitude more on software because the gains more than pay for themselves.

That concept will go on to marketing teams, ops teams, finance teams. Every department lead will have a dream stack in their mind that unlocks capacity they didn't have before. Engineers will go in as adult supervision to see if a custom tool is above or below a company's "vibe line". That's a made up term for whether the productivity gains of a custom tool are worth the building, hosting, securing and supporting.

Core engineering teams are getting smaller, but that won't mean a smaller demand for engineers.



Steve Gilles
Founder, Lookahead

§4

The benefits are real

92%

ship faster than a year ago

56%

call the increase large, not marginal

84%

saw boilerplate work collapse

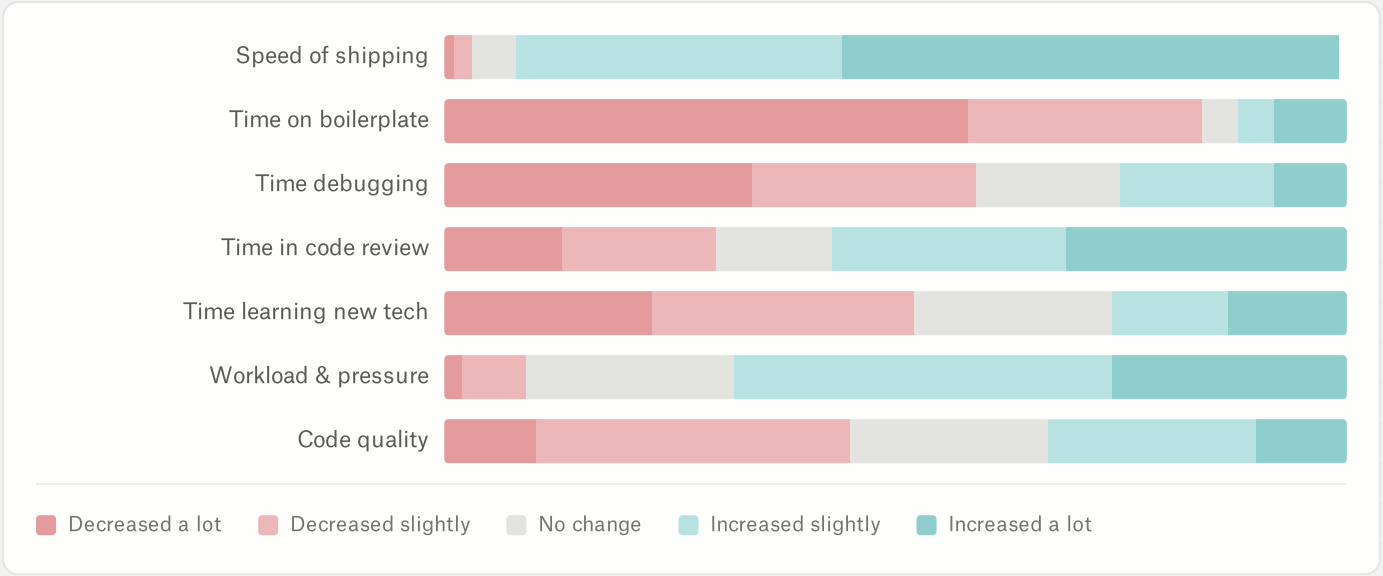
AI is compressing build times from months to days, delivering a massive increase in output that is fundamentally changing how we approach software engineering.

92% of respondents told us they ship faster than they did a year ago - and for more than half of everyone surveyed (55%), the increase was large rather than marginal. Only 3% have slowed down. This results are unsurprising. Alongside this increase in speed also was an increase in workload and pressure.

EXHIBIT 4.1

Speed and boilerplate moved decisively. Code quality is the one measure that went backwards.

Compared to 12 months ago, how has AI changed each of these? · n=247 each



n=247 · survey of 272 responses

Read: The first four lines are what everyone expected. The last one isn't - 45% report code quality decreasing against 33% reporting it improving.

Boilerplate code all decreased for 84% of respondents. What didn't is the work that was always hard: architecture, edge cases, and verifying correctness.

In our interviews the benefits were stark at times; one organisation measured three times the output per engineer per week. Another delivered a project scoped at two weeks in three days and a six-month project in two months. One engineering leader described diagnosing

an incident in seven minutes that would previously have taken four hours.

Ok but what does going faster mean?

When we asked people to name the single biggest benefit, speed came first - but the other answers underneath it describe something more specific than going faster.

EXHIBIT 4.2

The single biggest benefit is speed, and almost nobody says it is quality.

The single biggest benefit of AI in your work · single choice · n=272



n=272 · survey of 272 responses

Speed, freed-up time and less grunt work account for 69% of answers between them.

For many leaders speed is not the only benefit. For Pierre and Dave there is now an opportunity to place more bets on more things:

Another benefit mentioned repeatedly is that the barrier to entering an unfamiliar stack or domain has disappeared. Someone with no Android experience can now be productive in Android; that wall has largely vanished.

Does speed mean better?

Almost nobody is finding an increase in quality as a result of AI. We think that might change.

> "Leverage. A small team can now do what used to take a large one: the boring, repetitive work compresses to minutes, which frees people to spend their energy on the hard, interesting, creative problems."



Manik Surtani
CTO and Cofounder, Agentic AI Foundation

> "It's some multiple improvement — somewhere between 3x and 10x faster at building stuff."



Alex North
independent, formerly Google

"I expected that with AI, we'd be able to have a lot more stability in our delivery. One, it should take less time. Two, it should be quicker to do it. Three, it should be quicker to QA. Four, it should be more stable for the customer. We haven't found that, and I've struggled to figure out why."



Carl Woodward
CTO, Ordermentum

This indicates a classic bottleneck shift: speed without proportional review capacity transfers technical debt downstream. While commits are faster, defects are simply reaching production sooner.

The industry's most rigorous measurement effort found the same shape.

In the medium term I see quality increasing. Computers are getting better at writing code at an astonishing rate, refactoring is much easier, as is creating comprehensive test suites.

The best QA professionals are force multipliers, but they're rare. Sometimes called "developer in test", they sit as peers with developers, code well themselves, and set up automated tests that support the engineering team. They have the EQ to instill the belief that quality is everyone's job.

Problem is that when the budget for testing came at the expense of (perceived) velocity, that budget wasn't always allocated properly. We're now entering a world where creating a comprehensive suite of tests is table stakes.

The new problem is setting up your dev processes when there are more tests running against a even more PRs.



Steve Gilles
Founder, Lookahead

"If people are over-indexing AI as a tool? Do they have that core product capability, judgment, critical thinking? What are their inputs going into it? You cannot outsource that. Human in the loop is a bit of an overused phrase, but there has to be someone at the wheel at the moment."



Darren Smith
Senior Director Product Design, Multiplier

"A big part of my job as CTO - and for a while, leading product too - was saying no. We can't do it, it's too expensive, the ROI isn't there. Now the default answer isn't no. It's 'let me check."



Pierre Bergamin
CTO, InDebted

“It's not so much that more can happen. It's the things that wouldn't have happened previously can happen now... a refactoring here or a bug fix there... in the past [it] would have gone into the backlog and kept getting pushed down and eventually fallen off the bottom and never got done.”



Dave Slutzkin

Cofounder and CEO, Cadence

KEY IMPLICATION

The critical takeaway is that while output has scaled, it is not uniform. Productivity gains are currently concentrated in smaller teams. As the code base grows, engineering leaders must prioritize quality frameworks to prevent codebases from becoming unmanageable. Notably, non-technical teams report higher perceived quality improvements than engineering teams, suggesting a divergence in how "functional code" is defined.

The measure that moved most after speed was time in code review, which went up for 57%. AI has significantly decreased the cost of writing code but not the cost of checking it - so the hours saved at the keyboard are partly being repaid at the review stage, often by a human who still has to decide whether the machine's output is shippable.

§5

Key structural changes AI is driving

51%

of leaders are more hands-on

23%

picked “much more”

32 of 259

organisations have forward-deployed engineers

Smaller teams are the new baseline. In a down market, headcount is flat or falling, and leaders are staying technical to survive. When growth returns, that demand for leaders who are on the tools will still be there.



Steve Gilles
Founder, Lookahead

As shipping becomes cheaper and faster, the question of responsibility and accountability rises. Who does the work? What does ‘doing the work mean’? And what happens when things break? These emerging questions are pulling engineering teams in competing directions.

Leaders are closer to the tools

A year ago, senior leaders prided themselves on being hands-off. Today, many are back on the tools. Whether for fluency, prototyping or setting specifications. Practicing leadership has returned as a requirement for the role.

Note what Paul is describing: leaders practicing again - closer to the tools for fluency, prototypes and specs. Whether that extends to committing production code is a separate question, and it's where disagreement amongst leaders begins to emerge.

Of the 137 respondents who lead a team, 51% told us they've become more hands-on since adopting AI tools and 24% less. The extent is the interesting part: almost half of that 51% didn't pick "somewhat more". A full 23% of engineering leaders picked much more - almost as many as reported no change at all.

“I was not allowed to contribute to our codebase anymore probably six years ago - it had advanced too far beyond what I could do and I was just creating problems... Then GPT-3.5 came out and I started getting it to write code for me, and it was beautiful.* *All of my group engineering managers have gone back on the tools. There was a moment early this year when the harness got real good, then they jumped back on.”

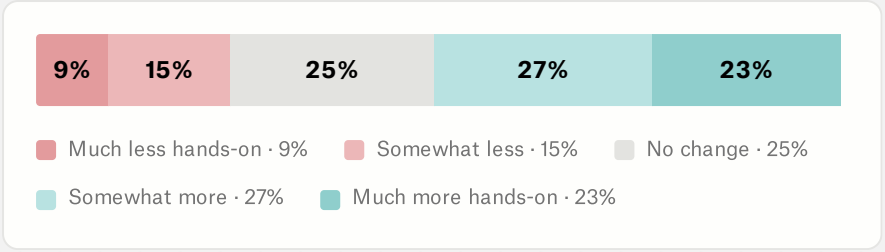


Dave Tong
Cofounder and CPTO, Employment Hero

EXHIBIT 5.1

More than half of engineering leaders are more hands-on, and the largest single group went all the way.

Since adopting AI tools, have you become more or less hands-on? · Engineering leaders only · n=132 of 137 eligible · mean 3.41 of 5



n=132 · of 137 eligible · survey of 272 responses

Almost as many leaders picked the strongest possible increase as picked no change at all.

So what does "being on the tools" actually mean? For some like Paul it means leading from the front. However some leaders offer a counter perspective to this.

For Wendy it is critical that whoever is writing production code retains responsibility and accountability for that code.

Alex North shared a similar view pointing to the practicalities of the situation: if code is cheap now, then having the most expensive person in the building producing it is a poor use of money and time.

Ray Tung, VP of Engineering at Kasada, had a slightly different concern. A senior leader who picks up real delivery work becomes a potential bottleneck because now critical time bound delivery is expected of the person with the least available time in the organisation. Routine, non-critical work is best delegated to junior engineers for skill development, but when delegation isn't possible, picking up these tasks yourself helps clear blockers and keep momentum going.

For Ciaran Hale, CTO at Deputy, being close to the work is less about personally producing every output and more about retaining the judgement to assess its quality. AI can make it easier to create something that looks convincing, but leaders still need enough context to distinguish a useful result from one that creates more work downstream.

For Chaitanya Kuber questions how much of this is happening inside companies at all.

> "12 months ago, I would have argued fiercely that if you're above 50 people, you shouldn't be anywhere near the tools... I think the biggest change over the last 12 months is you've seen the leaders all become practicing again."



Paul Keen
VP of Engineering, Culture Amp

> "I keep saying to people: on the tools is different to writing and committing production code."

> "If you are managing and responsible for outcomes across the board, if you're diving in deep, my question is: who is waking up at 3am if stuff goes wrong?"



Wendy Glasgow
CTO, HotDoc

“I personally don't know a lot of senior leaders who are returning to tools. I know a couple, but they're doing that in their own time — like, hey, I'm going to go build my own app or I'm going to go build my own startup. I'm seeing a lot more of that than 'I'm returning to the tool in my job, in my company.'”



Chaitanya Kuber
CTO, Airtasker

KEY IMPLICATION

The industry is clearly debating what hands-on should mean and how deep a leader should or shouldn't go. We are not going to settle this debate here but what isn't in doubt is that a clear pattern has emerged. More leaders are engaging directly with the work than a year ago.

Code is being written in the finance department, in the people team, in ops. Hands on might not mean shipping code all the time, but tech leaders responsible for setting up frameworks and guardrails that non-technical people can use to ship their own custom software securely with a stack that makes sense where costs can be controlled.

Tech teams will still build core products, but there will be an adult supervision department now too.



Steve Gilles
Founder, Lookahead

Mic Neale at Block has seen this first hand:

“Now the Product Requirements Document (PRD) is often a prototype instead of a document. Instead of debating for ages on a feature that might've taken a week or a month, you can spin something up in a few hours and share it with the team, or even push it live behind a feature flag for validation.”



Alexandra Swan
Software Engineer & Product Manager, Ajust

"[we] had a compliance workflow [that] takes three days. He's like wait I can actually build this all as skills that sit on top of an agent and run it from a Slackbot in a shared channel where people can see what's going on and approve things, so he did that. Some of it was a little bit janky but with a bit of help from an engineer was able to polish it up enough and put it in the right place. This is someone that has no direct engineering experience but boundless enthusiasm and curiosity from a totally different background. I saw it with people in finance and in legal doing that - just a lot of curiosity all over the place.

We can have challenges where things could accidentally leak out of the company because people are building things to scratch an itch because they realise once they get this tool, all the stuff they can do, and they're really proud of it. That's the other thing that surprised me, at least half the users were non-technical, and they were super proud of what they built. So they would go out of their way to show it to people."



Mic Neale
Principal Engineer Applied AI, Block

"Engineers need to become more product people and product people are going to become more engineers."

"One of my heads of product... is, that's his first time ever in his career, has deployed production code, you know. So that's a milestone for him and for me and for the company. Never before have I seen a product person do that."



James Brett
CTO Slatterys

Are we all builders now?

Yes! And this is a great thing. We're better recruiters because we've been in a dev team before and have empathy for the work. Some of the best product managers I know used to code. When everyone learns a bit about what it's like to write software, conversations about software start from a higher base.

What you should build and where you should stop is going to be the learning of 2027. People and teams need to find their "vibe line". How important is a custom tool, what sort of risk is it exposed to, and what's it cost to buy a roughly equivalent SaaS package?

Right now it seems like many are biting off more than they can chew, we'll soon see folks focusing more on their core business.



Steve Gilles
Founder, Lookahead

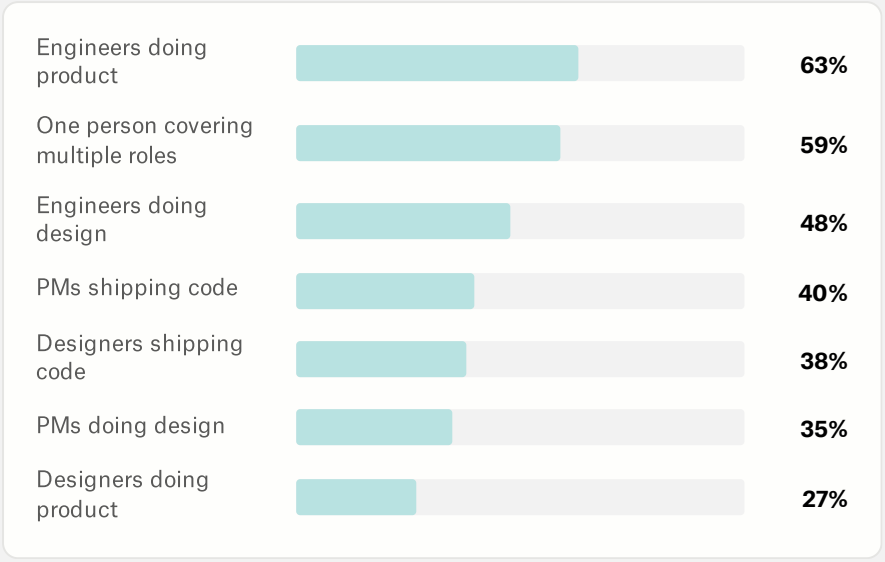
One of the most consequential things that happened this year is that non-engineers found out they can build. Not just spec it or wireframe it: build it. That has created friction but it has also broken down a barrier that sat between engineers and everyone else for decades. A working prototype turns out to be a far better way to communicate an idea than a document describing one.

Literally zero respondents told us they were seeing no cross-role movement. Non-engineers are shipping working prototypes as functional pull requests. Production ownership though remains squarely with Engineering for now but this might be up for debate.

EXHIBIT 5.2

Engineers take on product and design work far more often than product and design take on engineering.

Which cross-role shifts are you seeing? · Multi-select, so percentages sum above 100 · n=248 · 3.1 selections per respondent



n=248 · multi-select · survey of 272 responses

Engineering picks up work from its neighbours more often than it hands work back.

James Brett gave us some insight into where things could be heading:

Role boundaries are blurring, but ownership is not. While non-engineers can now generate functional prototypes, many in the industry remains firm on where accountability for production lies.

Carl Woodward, CTO at Ordermentum, watched this boundary resolve by itself. His product managers leaned into writing code, then pulled back once it dawned on them that shipping meant being on call. Ciaran Hale sees AI as expanding what product managers and designers can explore independently, particularly for early concepts and small improvements. But he is clear that this should not remove engineering judgement from more complex work, where requirements,

> "We've had this debate of like, you know, can product and design and other people write production code? And I'm like, hard no. I'm like ... you can build all of your prototypes, you can do all of that stuff, but unless you ... understand the complexity and the constraints, you cannot touch the [production] code."



Wendy Glasgow
CTO, HotDoc

> "There'll be this blurring and then the roles will solidify again."



Alex North
independent, formerly Google

architecture and long-term maintenance still matter. Vanessa Ung from Lorikeet questioned if a product built by generalists still works in three years.

Of the 22 leaders we interviewed, **only one expects this blurring to reverse.**

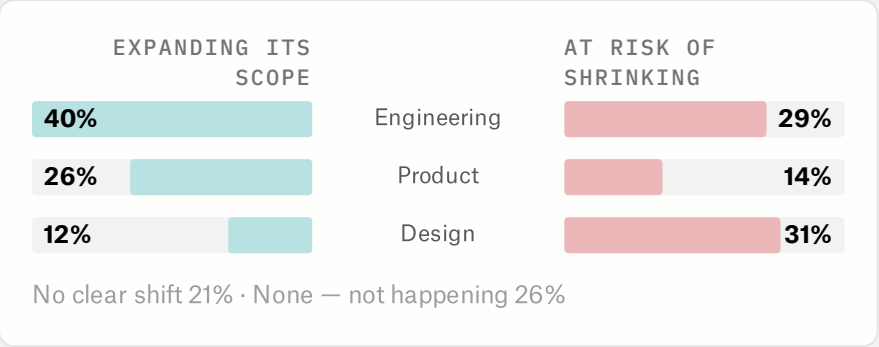
What about design?

Pierre Bergamin (CTO at InDebted) runs an engineering organisation with **no UX or UI designers at all**. The function was not downsized but it was removed completely. Paul Keen (VP Eng at Culture Amp) went the other way entirely and leaned hard into design: "we're finding designers have been the biggest unlock with AI tools. We found that they can pick up prototypes, they actually write pretty good code." John Allsopp would go further still: hire *more* designers, because engineering has stopped being the constraint.

Despite Paul and John's optimism our survey is pointing to the design function to be at risk (see caveat below).

EXHIBIT 5.3

Design is the only function named as at risk more often than as expanding.



n=272 · survey of 272 responses

Engineering and product are both named as growing more often than shrinking. Design runs the other way.

One caveat before you read too much into that. 233 of our 272 respondents work in engineering. Eleven are product managers. Fourteen are designers. Exhibit 5.3 tells you what engineers believe is happening to design, which is not the same as what designers are experiencing. We can't tell you the second, and we won't pretend otherwise.

> "Where I do see value is designers bringing their brain, being able to collaborate with other people, PMs, move upstream systems thinking, divergent thinking. AI can supplement that. And then [asking] how do we make sure we're bringing that design delight into the solutions that we're solving for as well?"



Darren Smith
Senior Director Product Design, Multiplier

KEY IMPLICATION

Building software is getting democratised. Ownership did not. Every leader we spoke with landed on the same boundary: engineering needs to maintain control over production. For those in Design many people see changes afoot that will redefine the role completely but it is up for debate whether this is a short-term experiment or a longer-term shift.

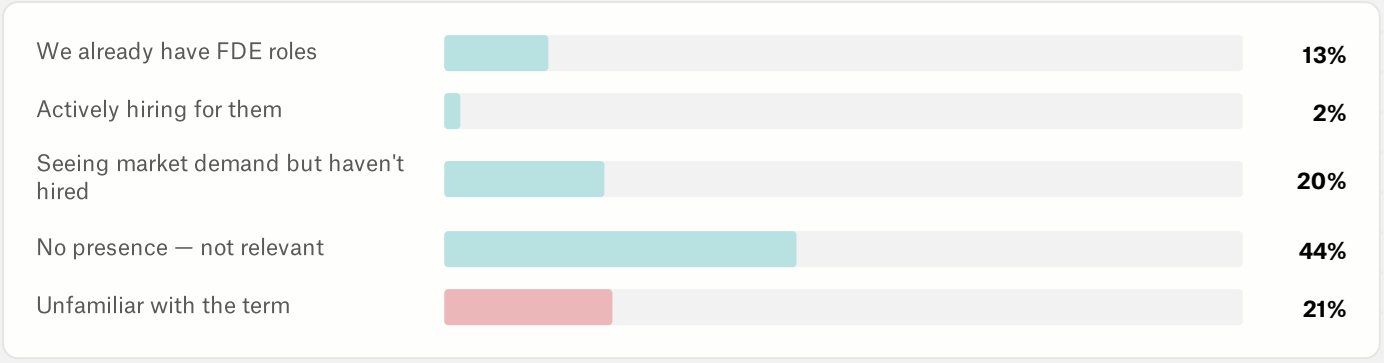
Forward Deployed Engineering: real or hype?

If you spend time on LinkedIn (like we do - it's a hazard of the job) you could be fooled into thinking the Forward Deployed Engineer (FDE) was the most sought-after role in 2026. The data tells a different story.

EXHIBIT 5.4

40 organisations in 270 have forward-deployed engineers. 57 respondents had never heard the term.

Which best describes forward-deployed engineering at your organisation today? · n=270 · one dot \= one respondent



n=270 · survey of 272 responses

The people who'd never encountered the term outnumber everyone who has the role and everyone hiring for it, combined.

48% claim a good understanding of the role but only 15% actually have the role.

Quite a number of people we spoke to were aware of the FDE role but some were sceptics. We heard that "it's a fancy word for integration or sales engineering", and still a "nothing thing". Some place it as a "solutions architect with a new badge from the AI labs". Some were blunter "they're salespeople".

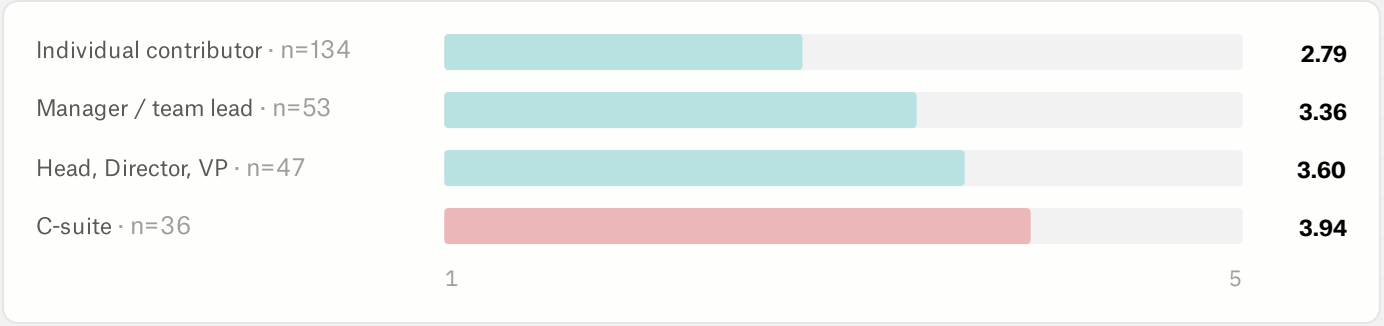
Not everyone is a sceptic. Manik Surtani sees the role as half engineer, half consultant, embedded with the customer and Pierre Bergamin has two dedicated FDEs running in-country model hosting for banks with data residency requirements.

Interestingly, a different level of appreciation for the role exists depending on your role.

EXHIBIT 5.5

Engineering leaders understand the term markedly better than the people they lead.

How well do you understand what a forward-deployed engineer is? · 1 \= no understanding, 5 \= very clear · n=270 · Cliff's δ 0.33 Solid q<0.001



n=270 · survey of 272 responses

One of the strongest results in the survey - it clears the strictest bar in both the seniority family and the manager-vs-IC family.

KEY IMPLICATION

Three numbers that shouldn't all be true at once.

48% say they have a good or very clear understanding of forward-deployed engineering

15% have the role

21% had never heard the term

Discourse is not demand and hype is not hired. It is a manager's role to learn about the trends that may reshape their companies so it is not surprising that managers are more aware of the role. This is a great illustration of an emerging role that is unlikely to be relevant to all organisations and for the sceptics out there just a rehash of something that already exists.

§6

Reshaping roles

$150–500

per engineer per month, the most common budget

14%

don't know their AI tooling budget

In some companies, hiring and AI tools draw on the same budget, so spending more on AI can mean less money for people. AI may also make more projects worth pursuing, giving companies a reason to invest more overall. As engineers, we need to understand how AI spending affects both hiring and what our teams can deliver.



Steve Gilles
Founder, Lookahead

As AI makes code faster and cheaper to produce, engineers bring more value through deciding what to build, how it should work and whether it's reliable. System design is becoming more important, but opinions on code review are divided: does it now demand more human judgement, or will AI take on that work too? There's also an unresolved question about how junior engineers will gain the experience to become senior engineers.

System design and product thinking matter more

We asked respondents to select up to three skills that had grown most in importance, and up to three that were becoming less important. System design led the gains at 56%, closely followed by product thinking at 53%.

"It used to lean heavily on how fast and fluently you wrote code. Now that's table stakes - the AI does plenty of it. The differentiator is judgment: knowing what to build, framing the problem well, spotting when the output is wrong, and reasoning about systems."

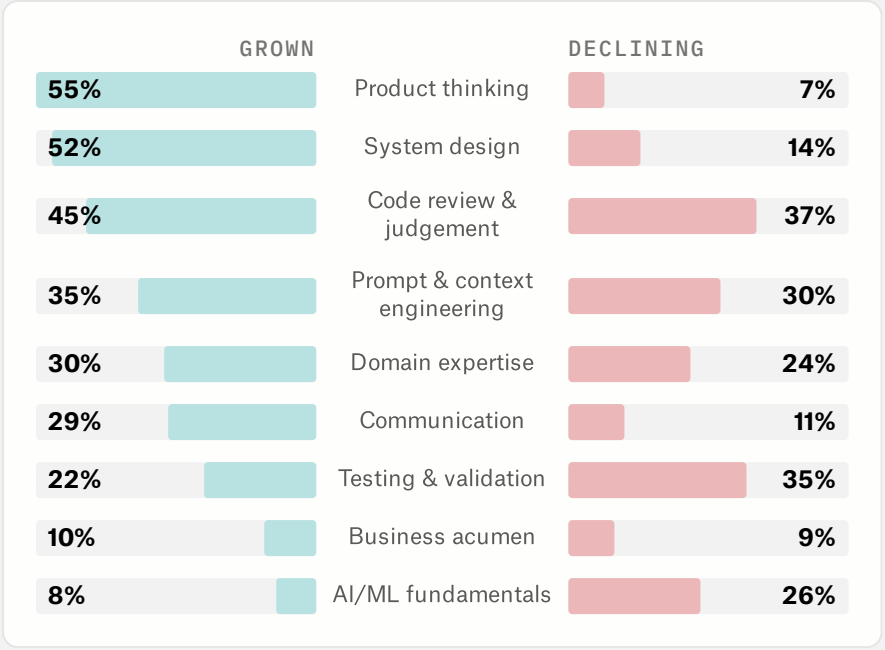


Manik Surtani
CTO and Cofounder, Agentic AI Foundation

EXHIBIT 6.1

System design and product thinking are settled. Code review and prompt engineering are contested.

Which skills have grown / are declining in importance? · pick up to 3 each · n=269 grown, n=246 declining



n=269 / 246 · survey of 272 responses

Read: Where a skill has a long bar on one side and a short bar on the other, the profession agrees. Where both bars are long, it doesn't.

Code review appears near the top of both lists. So does prompt engineering. At first glance this seems odd but we feel it represents two groups that are emerging. Each of these groups have reached opposite conclusions about the same job. One group believes that review is where human judgement is needed: the model produces 'the work', the engineer decides and deciding is now 'the work'. The other group believes that review itself is being automated — agents checking agents — and that prompt craft is a transitional skill, disappearing as models stop needing to be coaxed. The industry will undoubtedly shift one way, and our guess is the latter.

This begs the question, has our definition of what makes a great engineer changed at all? For our interviewees the response was mixed.

James Brett, however, suggests that the evolving nature of engineering work might leave some developers at a disadvantage.

One point of near-total agreement amongst those we interviewed: refusing to use AI is now disqualifying. Some described engineers not using AI as ridiculous, comparing it to refusing an IDE out of stubbornness or pride.

"I don't actually think it's changed anything — if anything, it enhances what we already knew to be true. Coding was never the most important part of being a software engineer."



Ray Tung
VP of Engineering, Kasada

"if someone's talented, they're talented. If you're looking for really really talented people, then I don't think a lot's changed."



Mic Neale
Principal Engineer, Applied AI, Block

"*Having AI as a prefix or suffix in a job title, I've basically outlawed. It's just the evolution of responsibilities and it should be part of your everyday work now. It should only be reserved for people working on the edge of that development.*."



Dave Tong
Cofounder and CPTO, Employment Hero

What about juniors?

This was the single most-raised topic. 17 of our 22 interviews brought it up unprompted but only **14% of respondents named erosion of junior skill development as the most significant downside of AI**. The routine tasks juniors used to cut their teeth on are now below what a model does unsupervised. As a result, the entry-level work is disappearing. The traditional pathway from graduate to senior which ran for years no longer has an obvious first step. Nobody is disputing this change.

From there the interviews divide. The pessimists worry that the on-ramp has gone and no replacement is in sight. James Brett went furthest, telling us he used to worry about the talent pool and no longer does — because he doesn't think the industry needs one coming in. Paul Keen's version is personal: his son is studying computer science, and he expects him to graduate and never write a line of production code.

The optimists though have a counter argument.

Early labour-market data gives weight to concerns about entry-level jobs. Stanford's Digital Economy Lab has been analysing US payroll data from ADP. Its August 2026 update found that employment among 22–25-year-olds in occupations highly exposed to AI was roughly 19% below where it would have been had it kept pace with peers in less-exposed occupations. That gap had widened from 15% a year earlier. Experienced workers showed no comparable gap.

The declines were concentrated in occupations where AI automates tasks rather than helps people do them, and appeared to reflect reduced hiring rather than increased job losses. That echoes what our interviewees describe: fewer opportunities to get started. But the researchers stress that these patterns do not prove AI caused the decline. We read them in the same spirit: an early warning, not a verdict.

This debate will not be settled in this report. The median respondent to our survey has 17 years in the industry, and 50% lead a team. Every account of the junior experience in this report is a seasoned person's account of it. That is worth noting.

Engineering gets a budget line?

Token spend arrived as a management question this year, and the survey suggests most teams are still working out what to do about it. **The modal budget is $150–$500 per engineer per month**, and 14% run uncapped but measured.

> "Back in the day, we used to be rewarded for single tasking... I think nowadays that's not true with agentic engineering. Good engineers are going to have to parallelize what they do. Fire something off, go to the next thing, fire something off, go to the next thing, come back. It's a bit like spinning plates, you know."



James Brett
CTO, Slatterys

> "Our ability to bring juniors through has all of a sudden become undefined. ... Those entering the workforce this year — what's their path to gaining experience? ... They studied for a world that is fast disappearing."

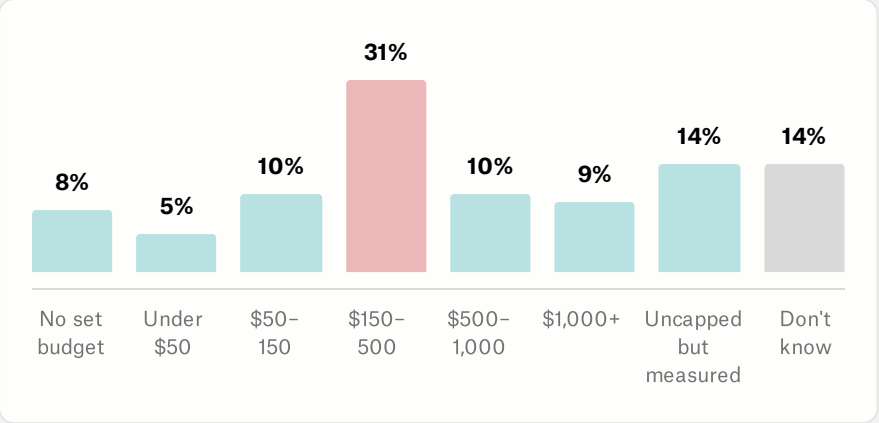


Chaitanya Kuber
CTO, Airtasker

EXHIBIT 6.2

$150–$500 per engineer per month is the most common budget, and 14% of people don't know theirs.

Roughly what is your monthly AI/LLM budget per engineer (AUD)? · n=272



n=272 · survey of 272 responses

Small companies spend more per head than large ones: 46% of respondents at 1–10 person companies are in the $150–500 band, against 12% at companies over 1,000.

For most of the leaders we interviewed, token budgets are a distraction and fundamentally the wrong management lever. Access to frontier AI models is now as essential as providing an engineer with a laptop; you don't ration the CPU cycles used for a build, and you shouldn't ration the tokens used for a solution. This represents a paradigm shift where the goal is thriving, not rationing. Instead of policing invoices, effective leaders are focusing on fostering a culture of continuous learning to ensure their teams stay relevant.

For Tim, framing AI as a budget line misses the point: it is a tool that expands what is possible, and we are only just beginning to map those new frontiers. If a leader finds themselves cutting AI budgets to save on costs, they have likely built the wrong culture. In this new reality, heavy token usage is not a cost to be managed, but a proxy for engagement and the necessary price of technical survival.

Between those two sits a governance camp running real instrumentation — quotas with peer-visible usage, prompt-quality and session-length metrics, hard subscription caps with no top-ups. Others are still reaching for the right unit of measurement altogether.

Underneath all of it sits a shared warning that has nothing to do with budgets.

One thing several interviewees converged on independently, and it matters more than the budget itself.

"I think junior engineers are going to learn faster because they don't have to write the code themselves."



Alex North
independent, formerly Google

"The junior is on the keyboard. The agent is asking the questions. The junior is talking to the senior about how they're going to answer those questions, and the amount of information that comes out of the senior's head in that process is actually more than ever would have in the past."



Dave Slutzkin
Cofounder and CEO, Cadence

"I don't believe in token budgets. Token budgets are bullshit."



Tim Lucas
Cofounder, Amp

"The people who use the most amount of tokens every month are not necessarily the people who are best at using AI."



Ray Tung
VP of Engineering, Kasada

Wendy Glasgow reaches the same conclusion from the opposite direction: a well-written prompt uses fewer tokens, so heavy usage can signal weak craft rather than strong output. Vanessa also cautions around the future cost of tokens. She says it will ultimately depend on your organisation and whether your core value proposition revolves around using AI.

"Given that tokens are subsidized effectively by the VCs, I'm not sure we've completely seen the real cost of those tokens yet. So, where does that land us? I don't know."

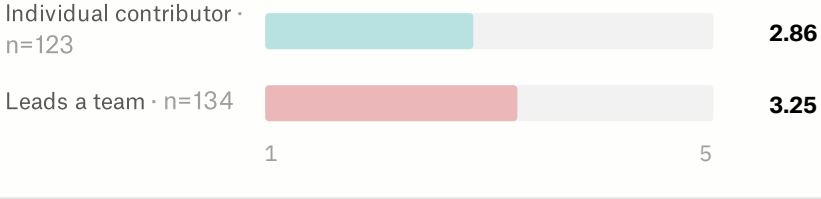


Vanessa Ung
Founding Engineer, Lorikeet

EXHIBIT 6.3

Leaders say token usage is carefully managed. The people they lead don't see it that way.

How carefully does your team manage AI/LLM token usage? · n=257 · Cliff's δ 0.18 Solid q \= 0.036



n=257 · survey of 272 responses

Clears the strictest statistical bar in the manager-vs-IC family.

"I think token efficiency is more of a term we're trying to get our heads around right now rather than token usage... if you're using the right model to answer the question, maybe that's the most efficient way even though it might be the most expensive."



Graham Darby
Data & AI Platform Manager, Tyro Payments

Code reviews are now a machine-to-machine interaction. The human role is shifting up the stack to intent and architecture. That might explain the opposing views in the survey - some view it as less important because humans will be less involved, others see it as more important because you're orchestrating agents so that's a bigger responsibility?



Steve Gilles
Founder, Lookahead

"AI adoption results in a token economics dilemma in a three-step journey. First, you get sold on the value and become locked into a platform. Second, usage grows ... it gets expensive as vendors move from bundled or per-user pricing towards premium token-based pricing. Third, you reach sufficient scale and costs skyrocket. At that point you have to ask whether it makes more sense to bring some of that capability in-house."



Shaun Domingo
CTO, Macquarie Cloud Services

KEY IMPLICATION

System design is named as grown more often than any other skill we tested - and it is the one that, two years ago, much of the industry was openly dismissing as interview theatre rather than a daily skill.

There is a clear contradiction about the importance of code review as a skill. The industry is split. People are looking at the same job and reaching opposite conclusions. This is likely to resolve itself one way or another over time.

On tokens, the gap between what leaders think they're managing and what their teams experience is one of the few things in this survey we can state as fact. Any organisation about to introduce quotas should assume its engineers do not currently know spend is being watched.

§7

Reshaping hiring

36%

of C-suite are hiring fewer engineers

11%

of managers say the same

1 in 25

engineering leaders is growing the team

I welcome this change. I have always advised companies to make their interview process an audition, and that's where we are now because hirers trust async code tests less.

So much time was wasted on boilerplate, and the level of care expected was unclear. Candidates usually worked on contrived examples after hours. It was hard to know whether they should build a lot roughly, or gold plate something small like it's production ready. Mostly it depended on who the reviewer was, and reviewers could vary within a company. We got to see the false negatives first hand.

Hiring isn't solved now though. There are more in person coding rounds, usually system design and running through real world examples. These are better but require more technical interviewer time, so there is pressure on candidates to perform to a "hell yeah" standard in early rounds. Hiring is still hard and I still probably have a job.



Steve Gilles
Founder, Lookahead

"Our interviews have the same starting point but where they go depends on the candidate. We ask follow up questions based on their answers and their real world experience so we see how they actually think through trade-offs rather than hearing an answer they had Claude write for them."



Ray Tung
VP of Engineering, Kasada

Nearly two in three organisations have changed how they interview, and the parts they changed are the parts that tested whether someone could write code. Underneath that, the market itself has tightened in a way that our respondents describe very differently depending on where they sit — and the people closest to the work are the ones still defending the coding test.

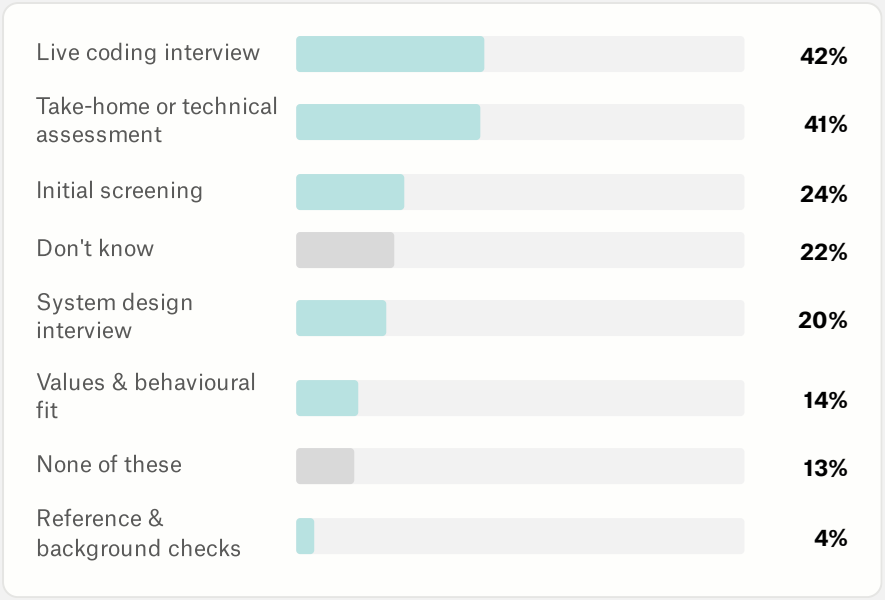
The interview loop got taken apart

60% told us their interview process has changed because of AI, with 24% describing the change as significant. When we asked which parts, the answer was unambiguous.

EXHIBIT 7.1

The take-home and the live coding interview absorbed almost all of the change.

Which parts of your hiring process have changed due to AI? · multi-select · n=258



n=258 · multi-select · survey of 272 responses

Values interviews and reference checks were largely left alone.

The dominant new pattern in our interviews is a practical exercise where AI is not merely allowed but expected, and the candidate is judged on how they frame the problem and critique what the model produced.

Several organisations described versions of the same thing: an hour with an unfamiliar codebase and your own tooling, where what's assessed is navigation and judgement rather than the finished solution. Paul Keen's team now lets candidates use an agent on the coding test and then asks them to explain what it wrote and why.

A significant camp has gone further and abolished coding tests outright.

And a stubborn minority has gone hard the other way - not out of nostalgia, but on a specific bet: that AI-assisted loops can be gamed, and that watching someone think, unassisted and in the room, is now the scarcer signal.

Others keep them for specific roles rather than abandoning them wholesale.

Our respondents are split almost exactly down the middle on whether coding tests remain relevant - 58% say yes, 42% say no. **What's more interesting is who says what.**

"You used to rely on all these tests, but you can just go and do the tests with agents now. So we've thrown all those out... [Now] I'd say, 'Here's your tool. Show me how you're going to solve this."



Dave Tong
Cofounder and CPTO, Employment Hero

"There's no code test because they're completely useless."



Dave Slutzkin
Cofounder and CEO, Cadence

"I think it's still very important to do coding interviews. I've gone very much back to the in-the-same-room, and I've even flown to different states to be in the same room as people."

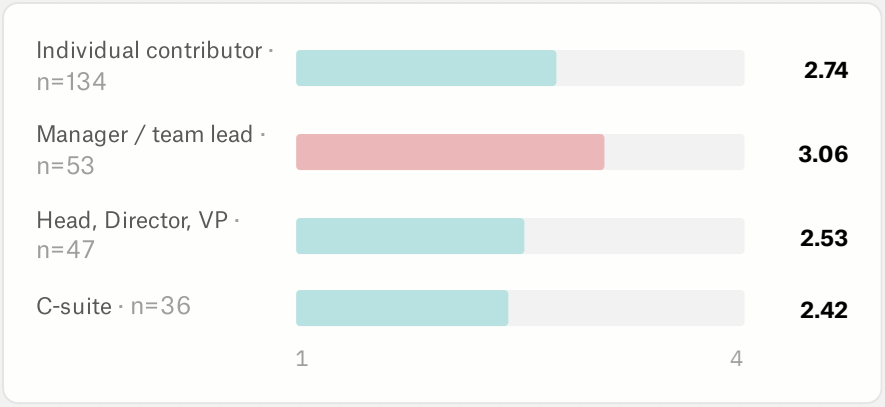


Gareth Stokes

EXHIBIT 7.2

Managers rate coding tests most relevant. The C-suite rates them least.

Are coding tests still relevant? · 1 \= not at all, 4 \= very · n=270 Strong signal q=0.061



n=270 · survey of 272 responses

The people who run hiring loops day to day are the most convinced they still work.

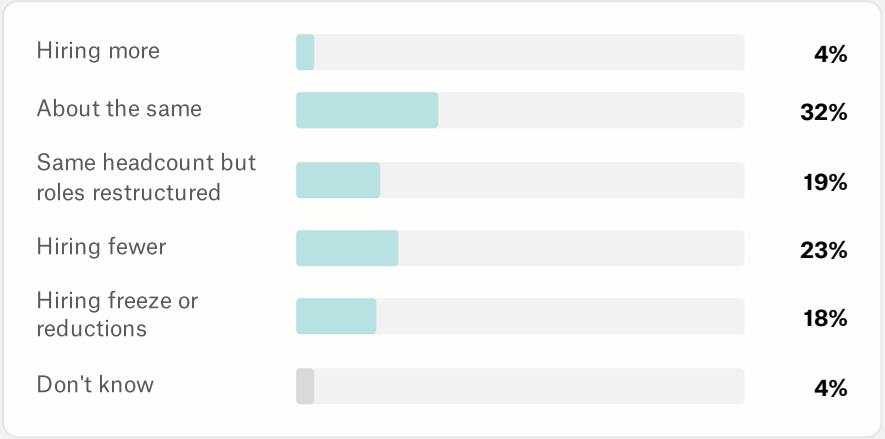
Who is actually hiring

Among the engineering leaders in our sample, the hiring market looks tight. **Only 4% are hiring more engineers.** A quarter are hiring fewer, and almost as many are holding headcount flat while restructuring what the roles are.

EXHIBIT 7.3

One engineering leader in twenty is growing the team.

Engineering headcount plan · engineering leaders only · n=125



n=125 · survey of 272 responses

Software engineering is still the hardest role to hire for, named by 34%, with engineering leadership second at 28% and AI engineering third at 27%.

The seniority split on this question is one of the largest effects anywhere in our data. **36% of C-suite respondents are hiring fewer**

"I would still do a coding test [for a data engineer] because they need to know capabilities of Python... they need to have good SQL skills, they need to know how to model data. Those things don't change."



Graham Darby
Data & AI Platform Manager, Tyro Payments

"The most valuable skill is still sound engineering judgement. AI can accelerate capable, curious engineers, but it does not replace the ability to frame a problem well, assess trade-offs and build something that will hold up in the real world."



Ciaran Hale
CTO, Deputy

engineers, against 12% of managers. Managers, in other words, mostly do not yet know.

What people are getting wrong in hiring came up repeatedly, and the most common answer was about role definition rather than candidate quality.

Where our interviewees do agree is on what to look for instead. The word that came up most was shipping.

This push for visible output creates a new hiring heuristic that is a fundamental misinterpretation of the shift. The trap for hirers is viewing AI as a reason to replace seasoned expertise with 'AI natives' or using it as a justification to reduce headcount based on raw speed. Instead, AI is a tool for learning and maintaining relevance. Don't ignore the reality that seasoned judgement remains the primary requirement for using these tools effectively.

What people are paid

The compensation picture is flatter than the productivity numbers would suggest. **57% of respondents told us their total package stayed the same over the past year.** 22% saw an increase, 6% a decrease, and 15% preferred not to say.

Where packages did move, they moved meaningfully: a median increase of $20,000, with a mean of $26,889 and a range running from −$30,000 to +$168,000. Increases were concentrated at larger companies: 40% of respondents at 201–1,000-person companies saw a rise, against 14% at companies of 1–10.

Separately, we asked engineering leaders what is happening to the premium their company will pay for exceptional engineers. **Not one respondent said it was falling significantly.** 37% said it is rising; 56% said no change.

57% said their total package stayed the same

+$20k median change among the 62 people whose pay moved

0% of engineering leaders say the premium for exceptional engineers is falling significantly

> "A lot of engineering leaders are hiring AI engineers. Everyone wants to hire an AI engineer but sometimes they don't really know why and what it is."



Pierre Bergamin
CTO, InDebted

> "Some people can just ship stuff and some people don't. Some people talk about shipping and there's a thousand reasons why things won't work. People like that get really frustrating. Always have been, but now it's extremely frustrating."



Mic Neale
Principal Engineer, Applied AI, Block

Market appetite for hiring engineers is not unlike my approach to alcohol now that I'm past 40 - less, but better.

Teams are getting smaller, clients are hiring one or two people at a time and we're seeing average salaries grow. In 2020 to 2022 the same engineer suddenly earned more and many scaleups were searching for 10+ people at a time. In 2026, companies are making a business case to hire fewer but more seasoned engineers. I worry what this means for non-senior engineers in the short term, and cover that later on.

Our experience is not indicative of the whole market. We're a niche agency, we're not the cheapest, and clients give us roles that are both critical and difficult. Volume hirers may have a different perspective.



Steve Gilles
Founder, Lookahead

Advice for people looking for work

Drawn from what our interviewees said they actually respond to.

Ship something, then show it. The single most common answer we got. Manik Surtani would rather see one small thing you actually built than a CV. Paul Keen wants a green-lit GitHub. James Brett sets the ceiling — five to ten shipped products and a real conversation about model trade-offs.

Spend real money on frontier tools. Treat it as a retraining cost, not a subscription. John Allsopp put it best: *"if you haven't been working for six months it might be an ask to spend $150 a month on tokens, but think of it like this: it's like five bucks a day, that's a cup of coffee a day. If you can find that money, develop those capabilities... build something, and be ambitious, right? You'd be astonished by what you can build."*

Bring an opinion, not a feature list. "Tried Cursor, Claude Code and Codex: here's why I landed where I did" is the new interview currency. Wendy Glasgow told us it's the exact thing she listens for.

Treat a gap as a reset, not a liability. Gareth Stokes doesn't care if you've been out six months. Tim Lucas is blunter: most of what mattered six months ago isn't relevant now anyway, so the field restarts more often than it used to — use the time to restart ahead of it.

"The default way that people do that is to say that younger candidates who are less exposed to the prior ways are better by default. "This is an opportunity for us to hire younger candidates who are AI native or whatever". That's wrong."



Oz Nova
founder, CS Primer

Advice for people hiring

Define the first ninety days before you write the job ad. The most common complaint we heard was undefined AI roles, not weak candidates - if your ninety-day description reads like a solutions architect, hire one of those instead.

Test the job as it actually exists. Interviewing without the tools people use daily is, as Oz Nova put it, a misrepresentation of the work. Judge problem framing and critique of the output, not typing.

Screen out the age shortcut. Assuming younger candidates are AI-native by default is a bias, not a strategy - Oz Nova was emphatic that it's simply wrong.

Recalibrate your gut on purpose. Dave Slutzkin's point: the person best suited to how development now works may not look like what you pictured, and gut feel was trained on the old job.

With product thinking growing in importance, hirers need to ask the right questions to draw the right evidence out. Hint: asking about times candidates had to make tradeoffs won't cut it.



Kyle Jackson
Technical Recruiter, Lookahead

KEY IMPLICATION

The coding test hasn't been killed, it's been split. And the split runs by seniority, with the people closest to the hiring loop defending it and the people furthest from it ready to drop it.

The headcount finding is the one to take seriously. Engineering leadership is the third-hardest role to fill while a quarter of leaders are hiring fewer people, and the C-suite knows something managers largely don't yet.

The compensation picture is the sharpest tension in the report. Output is up across almost every measure we tested, and the median engineer's pay did not move in our sample.

Prepare for interviews by building stuff. Opinion is split on whether it's best to build one polished thing or get through a bunch of side projects. And it probably doesn't matter: you do you. What's clear is that this is a time to make things. Fortunately, that's fun these days. Muck around with different tools and form an opinion.

A friend with one of the frontier model companies said they mostly screen for interest in keeping up with what AI can do. Keeping up is impossible but you need to be interested. If someone doesn't really care, that's an easy no. Given the pace of change, this makes sense.



Steve Gilles
Founder, Lookahead

§8

The downsides

68%

say workload and pressure went up

44 / 36

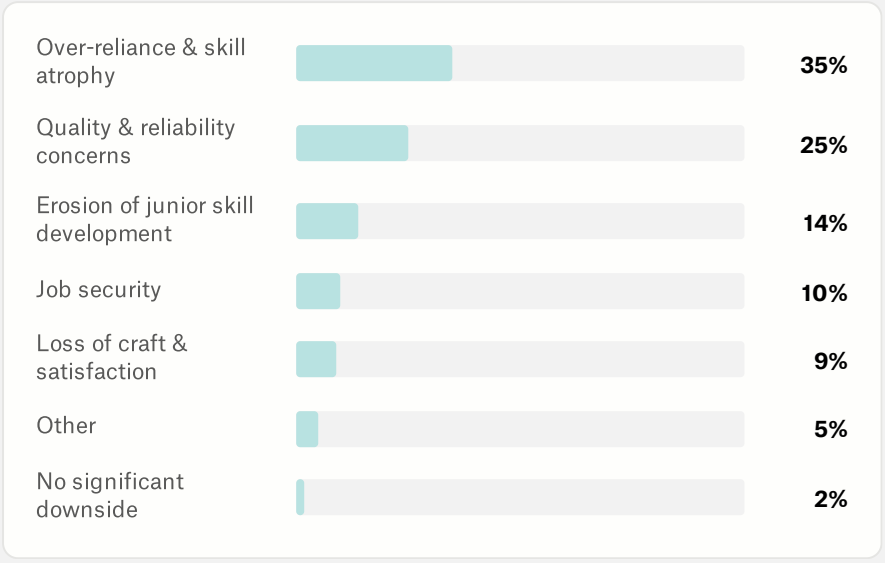
job satisfaction: down vs up

Public debate about AI often centres on job losses. Our respondents were more concerned about becoming too dependent on AI and losing their skills. **35% chose over-reliance and skill atrophy as AI's most significant downside**, making it the most common answer by a wide margin. Just 10% chose job security, one of the least-selected responses.

EXHIBIT 8.1

The biggest fear is forgetting how to do the work, not losing it.

The most significant downside or risk · single choice · n=272



n=272 · survey of 272 responses

Only 2% of respondents told us there is no significant downside at all.

Early research gives us reasons to take concerns about over-reliance seriously. In an MIT Media Lab experiment, 54 participants wrote essays using an LLM, a search engine or no tools. The LLM group showed the weakest brain connectivity measured by EEG and struggled to quote their own writing. The authors describe the potential cost of this reliance as "cognitive debt". It's a small preprint study of essay writing, so we should be cautious about applying its findings to engineering.

Closer to the workplace, Microsoft Research and Carnegie Mellon surveyed 319 knowledge workers about 936 examples of AI use.

"A major concern that I have is that people are turning their brains off when they use it, because you've kind of abdicated responsibility for verifying it yourself. We do have a bit more systemic risk, it feels like, from people just taking for granted some aspect of correctness."



Oz Nova
founder, CS Primer

"Big thing is - and I do this too - is there's no accountability anymore. So it's like 'Claude did this.' Cool, cool: I'm not paying Claude to do it. I'm paying you to do it."



Carl Woodward
CTO, Ordermentum

Greater confidence in AI was associated with less reported critical thinking. The study measured people's perceptions of their effort, rather than testing whether their skills had declined. Read the study.

Neither study proves that AI causes lasting skill loss. Both raise questions that echo the concern expressed by 35% of our respondents: are we losing practice in the skills we still need?

The workload paradox

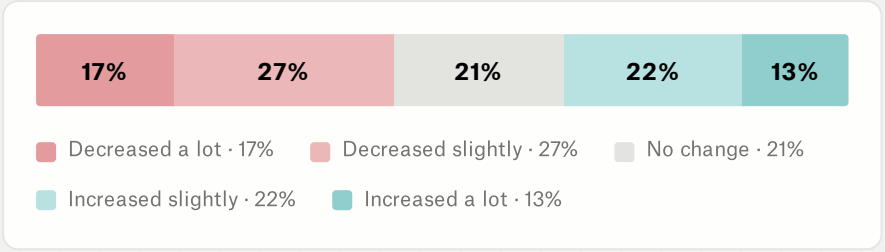
Here is the pairing that defines the year. Shipping got faster for 92% of people. Workload and pressure went up for 68%. And job satisfaction is the most divided question in the entire survey.

The mechanism our interviewees describe is cognitive, not organisational. The job used to reward hours of uninterrupted single-task flow. It now rewards supervising several parallel streams of machine output - checking, redirecting, context-switching - which is a different kind of tiring, and it taxes exactly the people who were best at the old mode.

EXHIBIT 8.2

Job satisfaction split almost exactly down the middle.

Compared to 12 months ago, how has AI changed your job satisfaction? · n=247 · mean 2.87 of 5



n=247 · survey of 272 responses

More people said satisfaction decreased a lot (17%) than said it increased a lot (13%).

Respondents with 15 or more years in the industry report higher job satisfaction than those with less.

"The drawback is cognitive overload. You can't let your agents blindly do everything - you still have to keep the reins on. Because I'm doing more and going deeper, that overload can get pretty extreme."



Alexandra Swan
Software Engineer & Product Manager, Ajust

"If I use it too much I walk away feeling frazzled. I'm running four to six problems at a time... I get this fatigue that I've not felt before, because my brain is just fried."

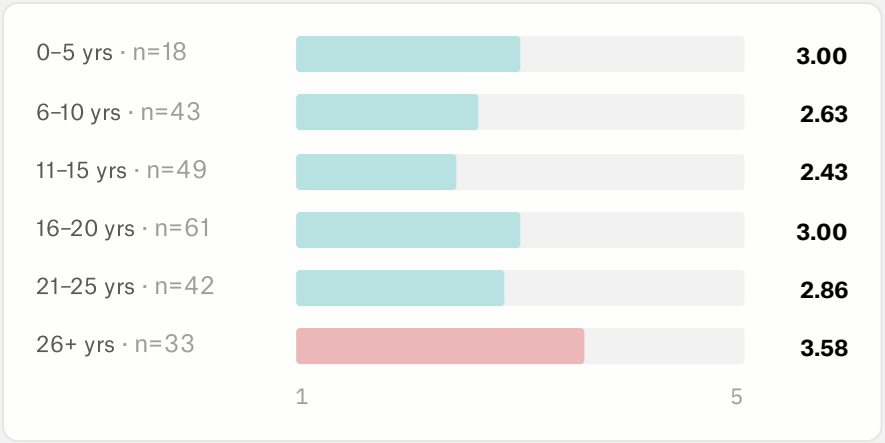


Dave Tong
Cofounder and CPTO, Employment Hero

EXHIBIT 8.3

Veterans appear to be enjoying this more than the people behind them.

Mean job satisfaction by years in industry · n=246 · ρ 0.16 Strong signal q=0.069



n=246 · survey of 272 responses

Honesty note: this pattern is graded as a Strong signal (q=0.069), a clear pattern rather than a proven fact. Workload runs the other way - newer engineers report higher workload and pressure - with the same grade (q=0.098).

What our survey missed

Three things came up across our interviews that we didn't specifically deal with in our survey but felt were important call outs.

The emotional cost of the transition. Around thirteen of our twenty-two interviews touched on it unprompted. This deserves to be called out.

Dave Slutzkin offered the most concrete version of why this lands unevenly: the job's cognitive profile has shifted from long uninterrupted flow to supervising six things at once, which disadvantages those who honed their craft through focus.

Everyone we spoke to loves the speed of prototyping. However many raised the question of what happens next?

Security specifically was raised by respondents as something we should have asked about and didn't. So was the gap between AI ambition and AI strategy.

Sitting under that strategy gap is a risk category that velocity metrics simply don't see: code shipping faster than anyone can security-review it, AI tooling adopted ahead of any policy on what data it touches, and agents operating with credentials nobody has audited. Our survey didn't ask, our respondents told us it should have, and we agree.

"I think there's an element of you're mourning this craft that you have developed your entire career over, and I think you've just got to acknowledge that craft has changed. The problem-solving is still there, and that's the benefit, but the craft is no longer— it's no longer there."



Paul Keen
VP of Engineering, Culture Amp

"Vibe coding works when you don't have customers. Works when you have a couple of customers. And vibe coding massively fails when you need to scale, you need to have any type of security or any type — and look, it will get better. ... But again, who's going to wake up and fix it when it breaks 3:00 in the morning?"



Wendy Glasgow
CTO, HotDoc

The environmental and social cost. Two interviewees raised it independently, and several survey respondents wrote it into their free-text answers.

> "Everyone's very conveniently ignoring the environmental cost of this. I don't know how that one plays out over the next 5, 10 years."



Chaitanya Kuber
CTO, Airtasker

Ciaran Hale also emphasised the need to be deliberate about where AI is used. The technology carries real infrastructure and energy costs, so the goal should not be more AI-generated output for its own sake, but better outcomes where the value clearly justifies the resource involved.

> Australia has lots of land and huge capacity for renewables. This race to build data centres plays to our strengths and I hope this opportunity is captured.



Steve Gilles
Founder, Lookahead

> "Everyone is seeking the AI promise, but many organisations don't have a clear strategy or practical plan for how they're going to implement it effectively. Good data is key."



Shaun Domingo
CTO, Macquarie Cloud Services

KEY IMPLICATION

The fear in this sample is one of deskilling, not replacement. Nearly four times as many people named skill atrophy as named job security. That is a different problem with different solutions, and most organisational responses to AI anxiety are aimed at the wrong one.

The workload finding is the uncomfortable one. Shipping got faster for almost everyone, workload went up for two thirds, and pay stayed flat for most. Whatever the speed bought, it does not appear to have been bought back by the people producing it.

And the emotional cost is real enough that thirteen of twenty-two interviews raised it without being asked. Our survey had no question for it, which is why it appears here in their words rather than in a chart.

§9 What's old is new again

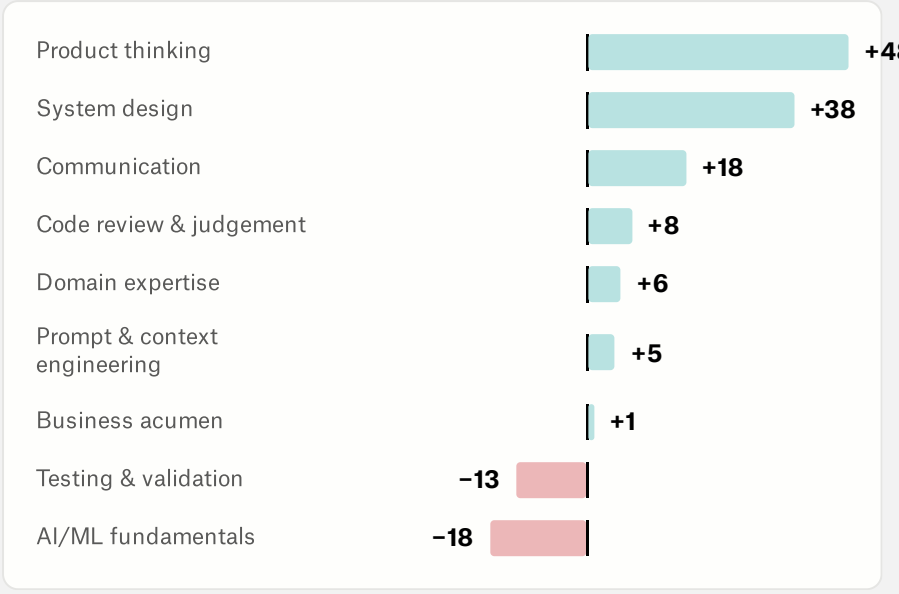
System design interviews have often felt like theatre: a performance that reveals little about how someone would design a real system. The frustration was with the interview, not the value of the skill.

Now, system design tops the list of skills our respondents say have grown in importance. As AI makes code easier to produce, deciding how the pieces fit together matters more. That renewed emphasis on design and judgement is one of our most encouraging findings in this report.

EXHIBIT 9.1

Ranked by net gain, the skills that grew are the ones that were always hard to teach.

Net score: % naming a skill as grown, minus % naming it as declining · n=269 and n=246



n=269 / 246 · survey of 272 responses

Read: Positive bars are skills the profession agrees have gained. AI/ML fundamentals fell furthest on net, with testing and validation close behind - the second of those is worth worrying about given how much more code is being produced.

The interviews get at why. When generating code stops being the constraint, the constraint moves to knowing what should be built, how it should be built and future implications of what was built. Unsurprisingly Product thinking (+48 net) and system design (+38) sit as clear skills that are rising in importance. In many ways this is not new but what it reflects is how people are spending their time and the blurring of lines between the once segregated role types.

> "I fundamentally believe that a really well-designed system is going to outperform really nicely written code any time."



Wendy Glasgow
CTO, HotDoc

> "The people who performed well before still perform really well today. That hasn't changed. ... Some of it is just tenure — they've been around long enough that they can smell the problems."



Carl Woodward
CTO, Ordermentum

One of our interviewees inverted a piece of advice the industry has given for two decades. The conventional wisdom was to go deep on one stack. Carl Woodward's argument is that libraries and frameworks become *more* important rather than less, because a well-designed library gives a model better material to work with - and that going wide now beats going deep.

A related observation came up several times: the collapse of stack specificity. Respondents told us the skills that really declined weren't on our list at all — they named coding itself, and knowledge of particular frameworks. If a model can pick up any stack in an afternoon, being the person who knows Rails deeply is worth less than being the person who knows what to build.

Is the forward-deployed engineer just a presales architect?

Several interviewees made the same observation about role titles from different angles. The forward-deployed engineer, on this reading, is a solutions architect or a sales engineer with a better name, and the pattern is a recurring one rather than a new one.

That doesn't make the role fake. It makes it old. And an old role coming back under a new name is exactly what this section is about.

> The data is a sampling of the vibe, but fails to predict the trend.
>
> We hired our first FDE in 2014 when Palantir came to Sydney, then didn't hear the term again until 2025.
>
> What an FDE actually does and what it’s called will differ between companies, but we will be seeing much more demand for this sort of role.
>
> 
>
> **Steve Gilles**
> Founder, Lookahead

KEY IMPLICATION

The strongest consensus in our entire survey is that the most durable skills are product thinking and system design. They sit clear of everything else: 55% and 52% of respondents say they grew in importance, with only 7% and 14% calling them declined.

The uncomfortable companion to that is testing and validation, one of the clearest net losers on the list. More code is being produced than ever by both technical and non-technical teams, by a process that is confidently wrong some of the time. As the skillset for checking as well as the dedicated role declines it raises the prospect of risk.

§10 Thank you from Lookahead

A heartfelt thank you to everyone who contributed to this report. We learned a lot making this and we hope you found it valuable too, even if it was just reassurance you're not alone.

Reach out with any feedback: hi@lookahead.com.au

Your pals at Lookahead (Debbie Teakle, Fiona Chan, Kyle Jackson, Sarah Jacob, Steve Gilles)

Small plug 🔌🤏

Every recruiter tells you they're different. Let us show you.

Lookahead is a technical recruitment agency helping Australia's best engineers find companies who care about building great software. We'll hire anyone in a software team: engineers, data and AI specialists, engineering leaders, product managers and designers. We come from technical backgrounds ourselves, and we've been building our database of Australia's technology community for 14 years.

We're reimagining what recruiting can be in an AI-first world. Around Christmas we got an LLM in front of our data to see if it would help us remember the great people we know. Now we have a developer on the team, we run our own stack, and part of our workflow is becoming a SaaS product. We send the best job invites in the world and care deeply about candidate experience. We're better at hiring because we're builders too.

§11

Appendix

1 · How we did it

The survey ran during 2026 and collected 286 submissions, of which 272 contained answers and are used throughout. The fourteen empty submissions are excluded from every figure in this report.

Percentages are of the people who answered each question, not of all 272 respondents, and every exhibit states its own n. Three things move that number. Drop-off: counts fall through the survey, from 272 on the demographics to fewer on the last sections. Routing: the manager block was shown only to the 137 respondents who lead a team, and those findings are reported as applying to engineering leaders rather than everyone. Multi-select: where marked, respondents chose several options, so percentages sum above 100%.

We also conducted 22 long-form interviews between June and August 2026. Quotations are lightly edited from transcripts. We've cut filler ("um", "just", "kind of"), false starts and repetition, and tidied grammar where speech doesn't survive on the page. We haven't changed anyone's meaning, added words they didn't say, or removed their qualifiers. Where a transcript was unclear, or where we've compressed someone's point, we've paraphrased in our own words rather than using quotation marks.

Who answered

ATTRIBUTE	BREAKDOWN
Seniority	Individual contributor 135 (50%) · Manager or team lead 53 (19%) · Head, Director, VP 48 (18%) · C-suite 36 (13%)
Primary function	Engineering 233 (86%) · Design & UX 14 (5%) · Product Management 11 (4%) · Other 11 (4%) · Data & ML 3 (1%)
Company size	51–200 76 (28%) · 1–10 59 (22%) · 1,000+ 51 (19%) · 201–1,000 43 (16%) · 11–50 42 (15%)
Sector	SaaS / product 119 (44%) · Fintech 38 (14%) · Other 32 (12%) · Consulting 27 (10%) · Healthcare 19 (7%) · Media / gaming 17 (6%) · E-commerce 14 (5%) · Government 6 (2%)
Location	NSW 185 (68%) · VIC 46 (17%) · QLD 14 (5%) · New Zealand 6 (2%) · Elsewhere 6 (2%) · SA 5 (2%) · WA 5 (2%) · TAS 3 (1%) · ACT 2 (1%)
Years in industry	median 17 · mean 16.8 · range 0–35
Leads a team	137 (50%)

Four things to hold in mind while reading

Sydney-weighted. 68% NSW, 17% VIC. This is not a national sample.

An engineering sample. 233 of 272 work in engineering; claims about product managers or designers as populations are not supportable.

A senior sample. Median 17 years' experience and 50% lead a team. The junior experience in this report is inferred by seniors, never measured.

A convenience sample. Fielded through Lookahead's network, so it is not random and we quote no margin of error. Findings are directional.

2 · How we analysed it

Beyond the question-by-question tallies, we ran an exploratory screen of 153 statistical tests across five ways of splitting the sample - seniority, company size, sector, function and tenure - with Benjamini–Hochberg correction applied across the whole family to control for the fact that testing 153 things will turn up false positives on its own.

Two further families were run separately: managers against non-managers (25 tests) and strength of feeling (14 tests). They re-use the same variables at a different grain, and folding them into the main screen would count the same evidence twice.

With the full sample in, 17 findings in the main screen clear the strict q<0.05 threshold. 44 of the 153 tests sit at p<0.05 where chance alone would produce about 8, and 15 sit at p<0.005 where chance alone would produce fewer than one, so the signal is real rather than an artefact of running many tests. We have graded every comparative finding in this report and shown the grade next to the exhibit.

GRADE	MEANING	HOW WE'VE WRITTEN IT
Solid	q < 0.05	Stated as fact.
Strong signal	q 0.05–0.10	Stated as a clear pattern, not a proven fact.
Suggestive	p < 0.05, q > 0.10	A line of enquiry. Never leads a section.

The Solid findings: in the main screen, the company-size gradient on overall impact (q=0.013), the seniority gradient on understanding forward-deployed engineering (q<0.001), and a seniority split on whether the interview process has changed (q=0.013) - though that last one is substantively unsurprising, driven almost entirely by individual contributors not being involved in hiring. In the manager-vs-IC family: leaders understand forward-deployed engineering markedly better than the people they lead (q<0.001), they report token usage being managed far more carefully than their teams experience (q=0.036), and the same interview-involvement split (q=0.003).

One methodological note worth stating plainly. Tenure and seniority move together (ρ 0.45), so reporting them as separate findings would double-count. When we tested tenure, seniority and company size in a single model, company size did most of the work (β −0.20, p=0.001), a smaller seniority effect held (β +0.17, p=0.008), and tenure added nothing (p=0.065).

3 · What we'd do differently next year

Split the hands-on question. It captured direction but not object, so we can say leaders are more hands-on without saying what they are hands-on with. Production work and exploratory work should be separate options.

Go deep on security. Respondents raised it as a missing option and it was one of the most-discussed topics in our interviews.

Ask more about the emotional cost. Thirteen of twenty-two interviews raised mourning, burnout or identity without being prompted. We had no question for any of it.

Reach more juniors, product managers and designers. A median of 17 years' experience and mostly engineers in our sample means we missed some learnings.